UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/885,456	06/19/2001	Kenneth J. Hines	10488/12:1	2742

7590 09/21/2005

MICHAEL R. BARRE
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025

EXAMINER	
YIGDALL, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 09/21/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/885,456	HINES, KENNETH J.	
	Examiner	Art Unit	
	Michael J. Yigdall	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 June 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *12 and 14-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *12 and 14-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's amendment and response filed on June 27, 2005 has been fully considered. Claims 12 and 14-29 remain pending.

Response to Arguments

2. Applicant's arguments have been fully considered but they are not persuasive.

Applicant contends that "neither Niemi nor Shindou discloses or suggests providing a simulation of a distributed software environment 'substantially simultaneously with execution of the ... software programs in the distributed software environment,' as recited in claim 12" (Applicant's remarks, page 9, last paragraph).

However, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

As presented in the previous Office action, Niemi teaches providing a representation of a distributed software environment (see, for example, FIG. 6 and column 13, lines 44-50). The representation is based on event records that are generated during the execution of software programs in the distributed software environment (see, for example, column 11, lines 1-9). The event records are logged to a file (see, for example, column 12, lines 7-9). Niemi expressly discloses that "the logging of error, trace, audit and other information can be enabled without having to close and restart the subject applications or processes" (column 15, lines 46-48). In other words, the event records are logged substantially simultaneously with the execution of the software programs, as Applicant acknowledges (Applicant's remarks, page 10, second full

paragraph). Indeed, Niemi further discloses that one may access and display the log file at any time (see, for example, column 13, lines 12-15). Niemi thus teaches providing a representation of a distributed software environment substantially simultaneously with the execution of the software programs in the distributed software environment.

Although the log file representation that Niemi provides substantially simultaneously with the execution of the software programs is not expressly a "simulation," Shindou teaches providing a simulation of a target system based on captured trace results (see, for example, FIG. 8 and column 10, lines 36-39). The trace results are event records (see, for example, FIG. 7 and column 9, lines 51-55). Shindou discloses that the simulation enables one to observe the instruction execution order and changes in the internal state of the target system in a synchronized manner (see, for example, column 11, lines 25-40).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Niemi to provide a simulation based on the event records generated and logged to the file during execution, such as taught by Shindou. The modification would have been obvious because one of ordinary skill in art would have been motivated to observe the changes in the state of the distributed software environment in a synchronized manner.

Therefore, Niemi in view of Shindou would teach providing a simulation of a distributed software environment substantially simultaneously with the execution of the software programs in the distributed software environment.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 12, 14-16 and 18-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,470,388 to Niemi et al. (art of record, "Niemi") in view of U.S. Pat. No. 6,145,099 to Shindou (art of record, "Shindou").

With respect to claim 12 (currently amended), Niemi discloses a method comprising:

(a) instantiating a distributed software environment that includes two or more physical processing elements and a runtime debugging architecture (see, for example, FIG. 2 and column 5, lines 21-26, which shows a distributed software environment having two or more workstations, i.e. physical processing elements, and FIG. 3 and column 6, lines 8-13, which shows elements of a runtime debugging architecture);

(b) executing first and second software programs in respective first and second physical processing elements within the distributed software environment (see, for example, column 6, lines 1-3, which shows executing first and second applications in the first and second workstations);

(c) generating event records during execution of the first and second software programs, in response to occurrences of events (see, for example, column 11, lines 1-9, which shows generating event records during execution in response to occurrences of events);

(d) receiving at least some of the event records at the runtime debugging architecture of the distributed software environment (see, for example, column 8, lines 21-35, which shows receiving the event records at the runtime debugging architecture); and

(e) forwarding at least some of the event records from the distributed software environment to a debugging host outside of the distributed software environment (see, for example, FIG. 2 and column 11, lines 30-38, which shows forwarding some of the event records to a centralized logging facility, i.e. a debugging host outside of the distributed software environment).

Although Niemi discloses providing a representation of the distributed software environment at the centralized logging facility or debugging host to show state changes based on the event records (see, for example, FIG. 6 and column 13, lines 44-50), Niemi does not expressly disclose:

(f) providing a simulation of the distributed software environment at the debugging host, wherein the simulation includes state changes based at least in part on one or more of the event records received from the distributed software environment.

However, Shindou discloses providing a simulation of a target system based on captured trace results (see, for example, FIG. 8 and column 10, lines 36-39). The trace results are event records (see, for example, FIG. 7 and column 9, lines 51-55). Shindou further discloses that the simulation enables one to observe the instruction execution order and the internal state changes of the target system in a synchronized manner (see, for example, column 11, lines 25-40).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the method of Niemi so as to use the event records received from the distributed software environment to provide a simulation at the debugging host, such as taught by Shindou. The modification would have been obvious because, for example, one of ordinary

skill in the art would have been motivated to observe, in a synchronized manner at the debugging host, changes in state of the distributed software environment.

Niemi in view of Shindou also discloses the limitation wherein the operation of providing a simulation of the distributed software environment comprises simulating the distributed software environment at the debugging host substantially simultaneously with execution of the first and second software programs in the distributed software environment (see, for example, column 15, lines 38-48, which shows providing the event records for the simulation concurrently with execution of the applications in the distributed software environment).

With respect to claim 14 (previously presented), the rejection of claim 12 is incorporated, and Niemi also discloses:

(a) collecting event records at a runtime system within the distributed software environment (see, for example, column 11, lines 1-9, which shows collecting event records at a runtime system within the distributed software environment);

(b) forwarding event records from the runtime system to the runtime debugging architecture (see, for example, column 10, lines 15-18, which shows forwarding event records to the runtime debugging architecture); and

(c) forwarding event records from the runtime debugging architecture to the debugging host along a communication channel (see, for example, column 10, lines 24-26 and column 11, lines 30-38, which shows forwarding event records to the centralized logging facility or debugging host along a communication channel).

With respect to claim 15 (previously presented), the rejection of claim 12 is incorporated, and Niemi also discloses the limitations wherein:

(a) the runtime debugging architecture adds time stamps to event records (see, for example, column 12, lines 16-19, which shows adding time stamps to event records); and

(b) the runtime debugging architecture adds causality stamps to event records, to identify causes of events associated with the event records (see, for example, column 12, lines 25-29, which shows adding causality stamps to event records to identify causes of events).

With respect to claim 16 (previously presented), the rejection of claim 12 is incorporated, and Niemi also discloses the limitation wherein the operation of forwarding event records to a debugging host comprises forwarding event records to the debugging host via an intermediate processing element (see, for example, column 12, lines 3-7, which shows forwarding event records by way of a logger or intermediate processing element).

With respect to claim 18 (previously presented), the rejection of claim 12 is incorporated, and Niemi also discloses the limitation wherein the operation of generating event records comprises generating an event record in response to at least one occurrence selected from the group consisting of:

(a) an occurrence of an event selected for logging according to a predetermined design model associated with the distributed software environment (see, for example, column 8, lines 21-35, which shows generating event records according to a predetermined design model such as from user input); and

(b) execution of an explicit event recording call included in one of the software programs (see, for example, column 10, lines 1-6, which shows generating event records upon execution of a debugging or event recording call included in the application).

With respect to claim 19 (previously presented), the rejection of claim 12 is incorporated, and Shindou further discloses:

(a) monitoring, with a probe, a bus trace associated with at least one of the physical processing elements (see, for example, column 7, lines 26-29 and column 8, lines 20-28, which shows a probe for monitoring a bus trace associated with the target system, i.e. one of the physical processing elements); and

(b) wherein the operation of generating event records comprises generating event records in response to activity detected on the bus trace (see, for example, column 9, lines 47-55, which shows generating event records in response to the bus trace).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the method of Niemi so as to implement the logging in hardware, as suggested by Niemi (see, for example, column 6, lines 61-65), by monitoring a bus trace associated with at least one of the physical processing elements with a probe, such as taught by Shindou. The modification would have been obvious because, for example, one of ordinary skill in the art would have been motivated to observe the internal signals of the physical processing element (see, for example, Shindou, column 8, lines 11-18).

With respect to claim 20 (previously presented), the rejection of claim 12 is incorporated, and Niemi also discloses:

(a) transmitting an event token from the distributed software environment to the debugging host (see, for example, column 11, lines 53-65, which shows transmitting an encapsulated event message, i.e. an event token, from the distributed software environment to the centralized logging facility or debugging host);

(b) expanding the event token at the debugging host, based at least in part on a predetermined sequence of events associated with the event token (see, for example, column 11, line 66 to column 12, line 3, which shows decapsulating or expanding the event token, and column 12, lines 7-9, which further shows extracting its information).

Shindou further discloses:

(c) simulating the distributed software environment, based at least in part on the predetermined sequence of events associated with the event token (see, for example, column 10, lines 51-59, which shows simulating the target system based on the predetermined sequence of events).

With respect to claim 21 (previously presented), the rejection of claim 12 is incorporated, and Niemi also discloses the limitation wherein the operation of providing a simulation of the distributed software environment comprises:

(a) tracking state changes that occur within a simulated environment in the debugging host, based at least in part on event records received from the distributed software environment (see, for example, FIG. 6 and column 13, lines 24-40, which shows tracking state changes that are to occur within the simulation, such as changes in error, trace and audit states, based on the event records from the distributed software environment).

Note that Shindou similarly discloses monitoring the state of the simulation (see, for example, internal state monitor 304 in FIG. 8).

With respect to claim 22 (currently amended), the claim recites an apparatus that corresponds to the method recited in claim 12 (see the rejection of claim 12 above).

With respect to claim 23 (previously presented), the claim recites an apparatus that corresponds to the method recited in claim 15 (see the rejection of claim 15 above).

With respect to claim 24 (currently amended), the claim recites a system that corresponds to the method recited in claims 12 and 14 (see the rejection of claims 12 and 14 above). Note that Niemi also discloses the limitation wherein the software programs, when executed, trigger events (see, for example, column 11, lines 19-23).

With respect to claim 25 (previously presented), the claim recites a system that corresponds to the method recited in claim 15 (see the rejection of claim 15 above).

With respect to claim 26 (previously presented), the claim recites a system that corresponds to the method recited in claim 18 (see the rejection of claim 18 above).

With respect to claim 27 (currently amended), the claim recites a system that corresponds to the method recited in claim 19 (see the rejection of claim 19 above).

With respect to claim 28 (currently amended), the claim recites a debugging host that corresponds to the method recited in claims 12 and 21 (see the rejection of claims 12 and 21 above).

With respect to claim 29 (previously presented), the claim recites a debugging host that corresponds to the method recited in claim 20 (see the rejection of claim 20 above).

5. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Niemi in view of Shindou, as applied to claim 16 above, and further in view of U.S. Pat. No. 6,125,392 to Labatte et al. (art of record, "Labatte").

With respect to claim 17 (previously presented), the rejection of claim 16 is incorporated.

Although Niemi discloses that the logger or intermediate processing element stores the event records in a file (see, for example, column 12, lines 7-9), and although Shindou likewise discloses that the event records are first stored in a trace memory of a probe unit, i.e. an intermediate processing element (see, for example, FIG. 6 and column 9, lines 31-34), Niemi in view of Shindou does not expressly disclose the limitation wherein the operation of forwarding event records to the debugging host via an intermediate processing element comprises storing one or more of the event records in a flash memory before forwarding the event records to the debugging host.

However, Labatte discloses storing an event log or event records in flash memory (see, for example, column 6, lines 42-44) in a persistent manner so as to be accessible by the low-level BIOS (see, for example, column 1, lines 13-18).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the method of Niemi in view of Shindou so as to store the event records in flash memory, such as taught by Labatte, for the purpose of providing the low-level BIOS with access to the same event records.

Conclusion

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MY

Michael J. Yigdall
Examiner
Art Unit 2192

mjy

ANTONY NGUYEN-BA
PRIMARY EXAMINER